Author

Name: Himadri Dixit
Roll No: 21f1006310
Email: 21f1006310@ds.study.iitm.ac.in

Description

Build a multi-user(admin/user) ticket booking application with different functionality for admin and users. Users can book tickets, rate and search for shows whereas admin can apply CRUD on both venues and shows.

Technologies used

1. Flask: Web framework
2. HTML/CSS: Rendering web pages
3. Jinja2: Template Engine
4. Bootstrap: Layout/Styling
5. SQLite: Database
6. SQLAlchemy: ORM
7. Flask-Restful: REST APIs
8. Flask-Security: Authentication
9. Flask-werkzeug: Security
10. YAML: API Spec
11. Python: Language

DB Schema Design

1. Table for:
    1. Venue: Details like name, place, capacity, shows running.
    2. Show: Details like name, venue, rating, tags, price, timing, tickets remaining.
    3. User: Details like email, password.
    4. Role: Available roles in the database.
    5. Role_users: Roles assigned to users.
    6. Booking: Details of booking like user, show, number of tickets, rating.
2. One to Many relationship between show and venue.
3. Roles are updated in the role_users table automatically at the time of registering.

API Design

I implemented:

1. 4 GET endpoints.
2. 3 POST endpoints.
3. 2 DELETE endpoints.
4. 2 PUT endpoints.

5. 1 PATCH endpoints.

I can

1. Create & Read user.
2. Create & Read admin.
3. Create& Read shows.
4. Update & Delete shows with confirmation.
5. Create, Read venues.
6. Update & Delete venues with confirmation.
7. Get history of bookings for a particular user.
8. Stop taking bookings if the capacity is reached for that show.
9. Display shows in latest added order for each venue.
10. Search for shows on the basis of name, tags, rating and location of the venue.

Architecture and Features

Project Structure:

1. Application folder contains:
    1. api.py where API initialization has been done along with the endpoints.
    2. config.py with all the configurations.
    3. controllers.py with all the controllers.
    4. model.py where database initialization has been done.
    5. resources.py with all classes for the API.
    6. security.py where flask security has been initialized.
2. All HTML files are present in templates folder.
3. Static folder contains favicon.ico.
4. Instance folder contains database.sqlite3.
5. Main application is running in app.py.
6. API.yaml, README.md and Project_Report.pdf are also included.

Features:

1. The application has different login and register page for user and admin.
2. User can book multiple tickets for multiple shows.
3. User can access their history of bookings and rate movies for which they've booked tickets.
4. User can search for shows on the basis of name, tags, rating and location preference.
5. User can see shows added in latest order.
6. Further bookings are not accepted in case of a houseful.
7. Admin can perform CRUD operations on shows and venues.
8. Confirmation is required for editing or deleting shows or venues.

Video

https://drive.google.com/file/d/1wmGMF1_NmbmQklTjNpTED5E7jNxOgKf4/view?usp=sharing